UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/054,544	01/18/2002	Joerg Jahnke	P-5513	7147

24209 7590 07/20/2006

GUNNISON MCKAY & HODGSON, LLP
1900 GARDEN ROAD
SUITE 220
MONTEREY, CA 93940

EXAMINER	
PANNALA, SATHYANARAYA R	
ART UNIT	PAPER NUMBER
2164	

DATE MAILED: 07/20/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/054,544	JAHNKE ET AL.	
	Examiner	Art Unit	
	Sathyanarayan Pannala	2164	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *3 April 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-33* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-33* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Applicant's Amendment filed on 4/03/2006 has been entered with claims 1, 12, 16, 26 and 30. Claims 1-33 are pending in this Office Action.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims 1-33 are rejected under 35 U.S.C. 102(e) as being anticipated by chau et al. (US Patent 6,721,727) hereinafter Chau.

4. As per independent claim 1, Chau teaches a method implemented on computer for processing XML documents. The data is stored in a data store connected to a computer. A main table is created having a column for storing a document and it has one or more elements or attributes. One or mode side tables are created to store one

or more elements or attributes. The side tables are used to locate the data in the main table. A query selects the data in the data storage device is retrieved into a work space and then one or more XML documents are created. The document object model tree is traversed to obtain information to retrieve relational data (col. 2-3, line 59-60, lines 62-67, lines 3-5 and 11-12).

Chau teaches the claimed step of "storing an element record for every element of said plurality of elements in an element table of said relational database so that said relational database includes a plurality of element records, wherein each element record includes a unique element ID, and an element data set" as XML enables storing entire XML documents into a database. The root_id in the application table is unique element ID and the user creates root_id as a primary key of the application table. When there is no primary key in the table, then XML system create a primary key as DXXROOT_ID and all side tables will have this key (Fig. 3, col. 6, lines 38-40; col. 18, line 67 to col. 19, line 1 and col. 17, lines 55-61).

Further, Chau teaches the claimed step of "storing an attribute record for every attribute of said plurality of attributes in an attribute table of said relational database so that said relational database includes a plurality of attribute records, wherein said attribute record comprises an attribute data set for one attribute and an element ID of an element to which the one attribute is assigned wherein said element table and said attribute table include content of said markup document and further wherein a new markup document having a same content as said markup document can be constructed by retrieving said element data set in each of said plurality of element records stored in

said element table of said relational database and by retrieving said attribute data set in each of said plurality of attribute records stored in said attribute table of said relational database" as the side tables 302, 304, 306 and 308 correspond to the attribute tables whereas Application table 300 correspond to element table. Side tables are dependent on the Application table and side tables also use the same root_id or the XML system created primary key DXXROOT_ID (Fig. 3, col. 18, line 67 to col. 19, line 1; col. 17, lines 61-63; col. 7, lines 38-39; col. 8, lines 22-24 and col. 24, lines 50-67).

5. As per dependent claim 2, Chau teaches the claimed step of "element data set includes character data" as the XML system allows storing the entire XML documents as column data in the application table (col. 8, lines 53-54).

6. As per dependent claim 3, Chau teaches the claimed step of "element data set contains a parent element ID" as the application table has the root_id as well as the side tables (Fig. 3, col. 17, lines 53-55).

7. As per dependent claim 4, Chau teaches the claimed step of "element data set contains a parent element ID" as the application table has the root_id as well as the side tables (Fig. 3, col. 17, lines 53-55).

8. As per dependent claims 5, Chau teaches the claimed step of "element data set includes an element name" as the user decides how XML document data is to be

accessed in a database. The user defines a document access definition (DAD) as an element (col. 12, lines 61-663).

9. As per dependent claim 6, Chau teaches the claimed step of "storing, for every unique element name of the plurality of elements, an element name record including an element name and a corresponding unique element name ID in an element name table of said relational database" as the column of the side table contains the value of a location path of the specified type. Name of the column is the alias name of the location path which identifies an element (Fig. 3, col. 13, lines 64-67).

10. As per dependent claim 7, Chau teaches the claimed step of "storing, for every unique attribute name of the plurality of attributes, an attribute name record including an attribute name and a corresponding unique attribute name ID in an attribute name table of said relational database" as the column of the side table contains the value of a location path of the specified type. Name of the column is the alias name of the location path which identifies an element or attribute (Fig. 3, col. 13, lines 64-67).

11. As per dependent claim 8, Chau teaches the claimed step of "attribute data set includes an attribute name" as the attribute is the name of an XML element and it is the tag name. This has the unique and it is adopted form XPTH (col. 15, lines 35-38).

12. As per dependent claim 9, Chau teaches the claimed step of "attribute data set includes an attribute value" as multiple-occurring element text or attribute value when generating XML documents (col. 14, lines 65-67).

13. As per dependent claim 10, Chau teaches the claimed step of "attribute data set includes an attribute value" as multiple-occurring element text or attribute value when generating XML documents (col. 14, lines 65-67).

14. As per dependent claim 11, Chau teaches the claimed step of "the markup document is an XML document" as extensible markup language is for creating XML documents (col. 2, lines 20-22).

15. As per independent claim 12, Chau teaches a method implemented on computer for processing XML documents. The data is stored in a data store connected to a computer. A main table is created having a column for storing a document and it has one or more elements or attributes. One or mode side tables are created to store one or more elements or attributes. The side tables are used to locate the data in the main table. A query selects the data in the data storage device is retrieved into a work space and then one or more XML documents are created. The document object model tree is traversed to obtain information to retrieve relational data (col. 2-3, line 59-60, lines 62-67, lines 3-5 and 11-12).

Chau teaches the claimed step of "storing an element record for every element of said plurality of elements in an element table of said relational database so that said relational database includes a plurality of element records, wherein each element record includes a unique element ID, and an element data set" as XML enables storing entire XML documents into a database. The root_id in the application table is unique element ID and the user creates root_id as a primary key of the application table. When there is no primary key in the table, then XML system create a primary key as DXXROOT_ID and all side tables will have this key (Fig. 3, col. 6, lines 38-40; col. 18, line 67 to col. 19, line 1 and col. 17, lines 55-61).

Further, Chau teaches the claimed step of "storing an attribute record for every attribute of said plurality of attributes in an attribute table of said relational database so that said relational database includes a plurality of attribute records, wherein said attribute record comprises an attribute data set for one attribute and an element ID of an element to which the one attribute is assigned" as the side tables 302, 304, 306 and 308 correspond to the attribute tables whereas Application table 300 correspond to element table. Side tables are dependent on the Application table and side tables also use the same root_id or the XML system created primary key DXXROOT_ID (col. 18, line 67 to col. 19, line 1 and col. 17, lines 61-63).

Further, Chau teaches the claimed step of "storing, for every unique element name of the plurality of elements, an element name record including an element name and a corresponding unique element name ID in an element name table of said relational database" as by storage, the XML system provides mechanisms for storing

and retrieving XML documents in relational database. The DB2 XML extender 200 takes an XML document 206 as the input, stores XML document 206 in DB2 210 either internally inside DB2 210 or externally on the files system as one or more XML files 208. Chau teaches two storage techniques and they are Xcolumn defines how to store and retrieve entire XML documents as column data of the XML user defined type and this method allows storing of elements and attribute values (Fig. 2, col. 5, lines 40-42; col. 6, lines 6-12; col. 7, lines 52-58; col. 8, lines 2-25 and col. 19, lines 28-36).

Finally, Chau teaches the claimed step of "storing, for every unique attribute name of the plurality of attributes, an attribute name record including an attribute name and a corresponding unique attribute name ID in an attribute name table of said relational database wherein said element table and said attribute table include content of said markup document and further wherein a new markup document having a same content as said markup document can be constructed by retrieving said element data set in each of said plurality of element records stored in said element table of said relational database and by retrieving said attribute data set in each of said plurality of attribute records stored in said attribute table of said relational database" as the relational side tables are created for indexing elements or attributes of documents stored in an XML column. Creating number of side tables is based on the understating of the DTD and XML documents, the application table 300 has a root_id in common with each side table302, 304, 306 and 308. Every side table will have a unique attribute for an order_tab side table has order_key as the primary key (Fig. 3, col. 13, lines 57-59;

col. 18, line 67 to col. 19, line 3; col. 17, lines 61-63; col. 7, lines 38-39; col. 8, lines 22-24 and col. 24, lines 50-67).

16. As per dependent claim 13, Chau teaches the claimed step of "element data set includes character data" as the XML system allows storing the entire XML documents as column data in the application table (col. 8, lines 53-54).

17. As per dependent claim 14, Chau teaches the claimed step of "element data set contains a parent element ID" as the application table has the root_id as well as the side tables (Fig. 3, col. 17, lines 53-55).

18. As per dependent claim 15, Chau teaches the claimed step of "element data set contains a parent element ID" as the application table has the root_id as well as the side tables (Fig. 3, col. 17, lines 53-55).

19. As per independent claim 16, Chau teaches a method implemented on computer for processing XML documents. The data is stored in a data store connected to a computer. A main table is created having a column for storing a document and it has one or more elements or attributes. One or mode side tables are created to store one or more elements or attributes. The side tables are used to locate the data in the main table. A query selects the data in the data storage device is retrieved into a work space and then one or more XML documents are created. The document object model tree is

traversed to obtain information to retrieve relational data (col. 2-3, line 59-60, lines 62-67, lines 3-5 and 11-12).

Chau teaches the claimed "an element table wherein said element table is configured to store a plurality of element records corresponding to a plurality of elements of a markup document so that said relational database includes a plurality of element records, and further wherein each element record includes an assigned element ID field and an element data set field" as XML enables storing entire XML documents into a database. The root_id in the application table is unique element ID and the user creates root_id as a primary key of the application table. When there is no primary key in the table, then XML system create a primary key as DXXROOT_ID and all side tables will have this key (Fig. 3, col. 6, lines 38-40; col. 18, line 67 to col. 19, line 1 and col. 17, lines 55-61).

Further, Chau teaches the claimed "an attribute table wherein said attribute table is configured to store a plurality of attribute records corresponding to a plurality of attributes of said markup document so that said relational database includes a plurality of attribute records, and further wherein each attribute data record includes an element ID field and an attribute data set wherein said element table and said attribute table include content of said markup document and further wherein a new markup document having a same content as said markup document can be constructed by retrieving said element data set in each of said plurality of element records stored in said element table of said relational database and by retrieving said attribute data set in each of said plurality of attribute records stored in said attribute table of said relational database" as

the side tables 302, 304, 306 and 308 correspond to the attribute tables whereas Application table 300 correspond to element table. Side tables are dependent on the Application table and side tables also use the same root_id or the XML system created primary key DXXROOT_ID (col. 18, line 67 to col. 19, line 1; col. 17, lines 61-63; col. 7, lines 38-39; col. 8, lines 22-24 and col. 24, lines 50-67).

20. As per dependent claim 17, Chau teaches the claimed "the element data set includes a character data field" as the XML system allows storing the entire XML documents as column data in the application table (col. 8, lines 53-54).

21. As per dependent claim 18, Chau teaches the claimed "the element data set includes a parent element ID field" as the application table has the root_id as well as the side tables (Fig. 3, col. 17, lines 53-55).

22. As per dependent claims 19, Chau teaches the claimed "the element data set includes an element number field" as the invoice_number is a primary key and examiner interpreted as an element number (Fig. 3, col. 19, lines 31-35).

23. As per dependent claim 20, Chau teaches the claimed "element data set includes an element name field" as the sales_person name is interpreted as element name (Fig. 3, col. 19, lines 30-31).

24. As per dependent claims 21, Chau teaches the claimed "the element data set comprises an element name ID field" as the invoice_number is a primary key and examiner interpreted as an element name ID or the sales_person ID column can created in the sales_tab table (Fig. 3, col. 19, lines 31-35).

25. As per dependent claim 22, Chau teaches the claimed "an element name table wherein said element name table is configured to store a plurality of element name records, and further wherein each element name record includes an element name ID field and a corresponding element name field" as the application table is the same as the element table (col. 8, lines 53-56).

26. As per dependent claim 23, Chau teaches the claimed "attribute data set includes an attribute name and an attribute value" as the term beginning with ATTLIST refer to attributes of an XML document as listed in the LineItem.dtd and relational tables created for indexing elements or attributes of documents stored in an XML column and the table is specified by name, type, path and etc. (col. 11, lines 61-62 and col. 13, line 57 to col. 14, line 6).

27. As per dependent claim 24, Chau teaches the claimed "attribute data set contains an attribute name ID" as side tables will have attribute ID, for example see the side_table order_tab has order_key, which is interpreted as an attribute name ID (Fig. 3, col. 21, lines 26-30).

28. As per dependent claim 25, Chau teaches the claimed "an attribute name table wherein said attribute name table is configured to store a plurality of attribute name records wherein each attribute name record includes an attribute name ID field and a corresponding attribute name field" as the side tables will be the same as attributes table and the could have an attribute name, attribute name ID and etc., for example see the side_table order_tab has order_key (Fig. 3, col. 21, lines 26-30).

29. As per independent claim 26, Chau teaches a method implemented on computer for processing XML documents. The data is stored in a data store connected to a computer. A main table is created having a column for storing a document and it has one or more elements or attributes. One or mode side tables are created to store one or more elements or attributes. The side tables are used to locate the data in the main table. A query selects the data in the data storage device is retrieved into a work space and then one or more XML documents are created. The document object model tree is traversed to obtain information to retrieve relational data (col. 2-3, line 59-60, lines 62-67, lines 3-5 and 11-12).

Chau teaches the claimed "storing an element record for every element of a plurality of elements of said markup document in an element table of said relational database so that said relational database includes a plurality of element records, wherein each element record includes a unique element ID, and an element data set" as XML enables storing entire XML documents into a database. The root_id in the

application table is unique element ID and the user creates root_id as a primary key of the application table. When there is no primary key in the table, then XML system create a primary key as DXXROOT_ID and all side tables will have this key (Fig. 3, col. 6, lines 38-40; col. 18, line 67 to col. 19, line 1 and col. 17, lines 55-61).

Further, Chau teaches the claimed "storing an attribute record for every attribute of a plurality of attributes of said markup document in an attribute table of said relational database so that said relational database includes a plurality of attribute records, wherein said attribute record comprises an attribute data set for one attribute and an element ID of an element to which the one attribute is assigned wherein said element table and said attribute table include content of said markup document and further wherein a new markup document having a same content as said markup document can be constructed by retrieving said element data set in each of said plurality of element records stored in said element table of said relational database and by retrieving said attribute data set in each of said plurality of attribute records stored in said attribute table of said relational database" as the side tables 302, 304, 306 and 308 correspond to the attribute tables whereas Application table 300 correspond to element table. Side tables are dependent on the Application table and side tables also use the same root_id or the XML system created primary key DXXROOT_ID (Fig. 3, col. 18, line 67 to col. 19, line 1; col. 17, lines 61-63; col. 7, lines 38-39; col. 8, lines 22-24 and col. 24, lines 50-67).

30. The computer program product of claim 27, Chau teaches the claimed "storing, for every unique element name of the plurality of elements, an element name record including an element name and a corresponding unique element name ID in an element name table of said relational database" as XML enables storing entire XML documents into a database. The root_id in the application table is unique element ID and the user creates root_id as a primary key of the application table. When there is no primary key in the table, then XML system create a primary key as DXXROOT_ID and all side tables will have this key (Fig. 3, col. 6, lines 38-40; col. 18, line 67 to col. 19, line 1 and col. 17, lines 55-61).

31. The computer program product of claim 28, Chau teaches the claimed "storing, for every unique attribute name of the plurality of attributes, an attribute name record including an attribute name and a corresponding unique attribute name ID in an attribute name table of said relational database" as the side tables 302, 304, 306 and 308 correspond to the attribute tables whereas Application table 300 correspond to element table. Side tables are dependent on the Application table and side tables also use the same root_id or the XML system created primary key DXXROOT_ID (col. 18, line 67 to col. 19, line 1 and col. 17, lines 61-63).

32. The computer program product of claim 29, Chau teaches the claimed "storing, for every unique attribute name of the plurality of attributes, an attribute name record including an attribute name and a corresponding unique attribute name ID in an attribute

name table of said relational database" as the side tables 302, 304, 306 and 308 correspond to the attribute tables whereas Application table 300 correspond to element table. Side tables are dependent on the Application table and side tables also use the same root_id or the XML system created primary key DXXROOT_ID (col. 18, line 67 to col. 19, line 1 and col. 17, lines 61-63).

33. As per independent claim 30, Chau teaches a method implemented on computer for processing XML documents. The data is stored in a data store connected to a computer. A main table is created having a column for storing a document and it has one or more elements or attributes. One or mode side tables are created to store one or more elements or attributes. The side tables are used to locate the data in the main table. A query selects the data in the data storage device is retrieved into a work space and then one or more XML documents are created. The document object model tree is traversed to obtain information to retrieve relational data (col. 2-3, line 59-60, lines 62-67, lines 3-5 and 11-12).

Chau teaches the claimed "a memory having stored therein a module for transferring data from a markup document into a relational database" as the stored procedures and modules are code organization to compose XML documents (Fig. 7, col. 42, lines 42-53).

Further, Chau teaches the claimed "a processor coupled to said memory wherein execution of said module" as a server computer 104 executing software and other

computer programs and to connect the server system 104 to data sources 106 (Fig. 1, col. 4, lines 3-5).

Further, Chau teaches the claimed "storing an element record for every element of a plurality of elements of said markup document in an element table of said relational database so that said relational database includes a plurality of element records, wherein each element record includes a unique element ID, and an element data set" as XML enables storing entire XML documents into a database. The root_id in the application table is unique element ID and the user creates root_id as a primary key of the application table. When there is no primary key in the table, then XML system create a primary key as DXXROOT_ID and all side tables will have this key (Fig. 3, col. 6, lines 38-40; col. 18, line 67 to col. 19, line 1 and col. 17, lines 55-61).

Finally, Chau teaches the claimed "storing an attribute record for every attribute of a plurality of attributes of said markup document in an attribute table of said relational database so that said relational database includes a plurality of attribute records, wherein said attribute record comprises an attribute data set for one attribute and an element ID of an element to which the one attribute is assigned wherein said element table and said attribute table include content of said markup document and further wherein a new markup document having a same content as said markup document can be constructed by retrieving said element data set in each of said plurality of element records stored in said element table of said relational database and by retrieving said attribute data set in each of said plurality of attribute records stored in said attribute table of said relational database" as the side tables 302, 304, 306 and 308 correspond

to the attribute tables whereas Application table 300 correspond to element table. Side tables are dependent on the Application table and side tables also use the same root_id or the XML system created primary key DXXROOT_ID (col. 18, line 67 to col. 19, line 1; col. 17, lines 61-63; col. 7, lines 38-39; col. 8, lines 22-24 and col. 24, lines 50-67).

34. As per dependent claim 31, Chau teaches the claimed "storing, for every unique element name of the plurality of elements, an element name record including an element name and a corresponding unique element name ID in an element name table of said relational database" as XML enables storing entire XML documents into a database. The root_id in the application table is unique element ID and the user creates root_id as a primary key of the application table. When there is no primary key in the table, then XML system create a primary key as DXXROOT_ID and all side tables will have this key (Fig. 3, col. 6, lines 38-40; col. 18, line 67 to col. 19, line 1 and col. 17, lines 55-61).

35. As per dependent claim 32, Chau teaches the claimed "storing, for every unique attribute name of the plurality of attributes, an attribute name record including an attribute name and a corresponding unique attribute name ID in an attribute name table of said relational database" as the side tables 302, 304, 306 and 308 correspond to the attribute tables whereas Application table 300 correspond to element table. Side tables are dependent on the Application table and side tables also use the same root_id or the

XML system created primary key DXXROOT_ID (col. 18, line 67 to col. 19, line 1 and col. 17, lines 61-63).

36. As per dependent claim 33, Chau teaches the claimed "storing, for every unique attribute name of the plurality of attributes, an attribute name record including an attribute name and a corresponding unique attribute name ID in an attribute name table of said relational database" as the side tables 302, 304, 306 and 308 correspond to the attribute tables whereas Application table 300 correspond to element table. Side tables are dependent on the Application table and side tables also use the same root_id or the XML system created primary key DXXROOT_ID (col. 18, line 67 to col. 19, line 1 and col. 17, lines 61-63).

Response to Arguments

37. Applicant's arguments filed on 4/03/2006 have been fully considered but they are not persuasive and details as follows:

a) Applicant's stated as "Applicants have amended each of the independent claims to more clearly recite the invention."

In response to the Applicant's statement, Examiner respectfully informs that the Applicants Amendment made it clear and the rejection under 35 USC 112, second paragraph is withdrawn.

b) Applicant's argument stated as "Chau fails to teach or suggest: wherein said element table and said attribute table include content of said markup

document and further wherein a new markup document having a same content as said markup document can be constructed by retrieving said element data set in each of said plurality of element records stored in said element table of said relational database and by retrieving said attribute data set in each of said plurality of attribute records stored in said attribute table of said relational database."

In response to the Applicant's argument, Examiner respectfully disagrees, because Chau do teach the amended segment as listed above as For traditional SQL data, that is either decomposed form incoming XML documents or in existing relational tables to be used to create outgoing XML documents, the XML system provides a custom mapping mechanism (col. 7, lines 37-42). The XML system provides powerful user-defined functions (UDFs) to store and retrieve XML documents in XML columns, as well as to extract XML element/attribute values (col. 8, lines 22-24). Searching uses indices created on side table columns, which contain XML element contents or attribute values extracted form XML documents and with the default view, or any view created by the application, a user can search XML documents by a query on the side tables (Fig. 3, col. 24, lines 50-67). Applicant's arguments do not comply with 37 CFR 1.111(c) because they do not clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. Further, they do not show how the amendments avoid such references or objections.

Further, in response to applicant's argument, the claimed invention must result in a structural difference between the claimed invention and the prior art in order to patentably distinguish the claimed invention from the prior art. If the prior art structure is capable of performing the intended use, then it meets the claim.

Conclusion

38. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sathyanarayan Pannala whose telephone number is (571) 272-4115. The examiner can normally be reached on 8:00 am - 5:00 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Charles Rones can be reached on (571) 272-4085. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Sathyanarayan Pannala
Examiner
Art Unit 2164

srp
July 16, 2006